**Software Requirements**

**Specification**

**For**

**Farmer Social Network**

**Version 1.0**

**Prepared by Om & Vanraj**

**Dharmsinh Desai University of India December 2024**



|  |
| --- |
| Dharmsinh Desai University |
| Faculty of Technology, College Road,  Nadiad – 387001, Gujarat |

Table of Contents

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms, and Abbreviations
   4. Overview
2. Overall Description
   1. User Needs
   2. System Features
3. Functional Requirements
   1. User Authentication
   2. Post Management
   3. Comment Management
   4. Resource Management
   5. Notifications
   6. Admin Management
4. Non-Functional Requirement
   1. Performance
   2. Security
   3. Usability
   4. Scalability
   5. Payment Gateway Integration
   6. User Profile Management
   7. Manage Attraction Information
   8. Booking management
   9. Request to Add attraction
5. System Design
   1. Architecture
   2. Use Case Diagram

**1. Introduction**

**1.1** Purpose

The purpose of this document is to outline the functional and non-functional requirements for the Farmer Social Network Web Application. The system aims to provide a seamless way for Farmer And Agronomist interact with the platform. The functionalities include user authentication, Post and Discussion, Crop Treatment Resources , Real-time Chat and Notifications.Also Farmer And Agronomist can share their thoughts about the Crops treatement and also share their experience on particular topic related to Crops.Here, Farmer can get the suggestion from Agronomist for particular Crops treatement and other farming related topic.It will help to spread the Farmer interaction and help to give better result in Farming Sector.

**1.2** Scope

This document describes the scope for Farmer Social Network Web Application System.Here We have the three type of user (i)Farmer (ii)Agronomist (iii)Admin.Farmers can Create an Account.they Will post and Discussion on the relative topic related to farming.They can get the help from Agronomist on relative farming process or Treatement. Also they can have Real-time Chat with eachothers.Agronomists will also provide the help platform for Farmer to enhance their Cultivation Process.Also Admin Can handle user activities.

**1.3** Definitions, Acronyms, and Abbreviations

* Farmer: A user can get their Relative information via this Plateform.
* Agronomist: They can gave their idea and feedback to the Farmer For better Result in Farming.
* Admin: A system administrator responsible for managing users and system settings.
* **Post**: A user-created content piece to share knowledge or information.
* **Comment**: A response or feedback provided by users on a post.
* **Resource**: A file or document shared by users related to farming.

**1.4** Overview

This SRS document provides a detailed description of the functionalities, system design, and technical requirements for the Farmer Social Network.

**2. Overall Description**

**2.1** User Needs

* Farmers post and Discussion on the relative topic related to farming.
* Agronomists will also provide the help platform for Farmer to enhance their Cultivation Process.
* Admins need to manage all users (patients, doctors, receptionists) and perform system-wide tasks.

**2.2** System Features

* User Authentication:
  + Allow farmers and Agronomist to sign up and log in.
  + Implement roles (e.g., Farmer, Agronomist).
* Post and Discussion:
  + Enable users to share posts about cultivation techniques.
  + Allow others to comment and react to the posts.
* Crop Treatment Resources:
  + Provide a repository for crop treatment techniques.
  + Allow users to upload and download resources.
* Real-time Chat:
  + Implement chat functionality for farmers to connect and discuss.
* Notifications:
  + Notify users about new posts, comments, or direct messages
* Admin Management:
  + Suspend or delete user accounts if needed.
  + Take care about user can’t Post about any restricted content.

**3. Functional Requirements**

**3.1** User Authentication

* **Sign Up/Login**
  + **R.3.1.1: Create Account**
    - **Description:** Allows users to create a new account on the platform.
    - **Input:** User provides email, password, mobile number, and address.
    - **Output:** Successful account creation, enabling the user to log in.
  + **R.3.1.2: Login**
    - **Description:** Enables users to securely access their accounts.
    - **Input:** Username and password.
    - **Output:** Successful login, authenticating the user.

**3.2** Post Management

* **Post Management**
  + **R.3.2.1: Create Post**
    - **Description:** Allows users to create new posts to share knowledge or seek advice.
    - **Input:** User provides a title and content for the post.
    - **Output:** Post is created and saved to the database.
  + **R.3.2.2: View Posts**
    - **Description:** Enables users to browse existing posts on the platform.
    - **Input:** Request for posts.
    - **Output:** List of posts with title, content, author, creation date, and associated comments.
  + **R.3.2.3: Edit Post**
    - **Description:** Allows users to modify their previously created posts.
    - **Input:** Updated title or content of the post.
    - **Output:** Post is updated successfully.
  + **R.3.2.4: Delete Post**
    - **Description:** Provides users with the ability to remove their posts.
    - **Input:** Request to delete a specific post.
    - **Output:** Post is deleted from the database.

**3.3** Comment Management

* **Comment Management**
  + **R.3.3.1: Add Comment**
    - **Description:** Enables users to add comments on existing posts.
    - **Input:** User provides content for a comment on a post.
    - **Output:** Comment is added to the post.
  + **R.3.3.2: View Comments**
    - **Description:** Allows users to view all comments on a specific post.
    - **Input:** Request for comments on a specific post.
    - **Output:** List of comments on the post.

**3.4** Resource Management

* **Resource Sharing**
  + **R.3.4.1: Upload Resource**
    - **Description:** Enables users to share files and documents related to farming.
    - **Input:** User provides a title, description, and file for upload.
    - **Output:** Resource is saved and accessible to other users.
  + **R.3.4.2: View Resources**
    - **Description:** Allows users to view and access shared resources.
    - **Input:** Request for resources.
    - **Output:** List of uploaded resources with titles and descriptions.

**3.5** Notifications

* **Notifications**
  + **R.3.5.1: Comment Notifications**
    - **Description:** Alerts users when someone comments on their posts.
    - **Output:** User receives a notification.

**3.6** Admin Management

* **Admin Management**
  + **R.3.6.1: User Management**
    - **Description:** Allows admins to manage user accounts, including viewing, editing, or deleting them.
    - **Input:** Request to view, edit, or delete a user.
    - **Output:** User information is displayed, updated, or removed as per the request.
  + **R.3.6.2: Post Moderation**
    - **Description:** Enables admins to review and moderate posts for compliance with platform rules.
    - **Input:** Request to delete or flag inappropriate posts.
    - **Output:** Post is removed or flagged.
  + **R.3.6.3: Resource Management**
    - **Description:** Allows admins to review, approve, or delete shared resources.
    - **Input:** Request to review, approve, or delete a resource.
    - **Output:** Resource is approved, flagged, or removed.

**4. Non-Functional Requirements**

**4.1** Performance

* The system should handle up to 1,000 concurrent users.
* API response time should be under 2 seconds.

**4.2** Security

* Use encrypted storage for sensitive data (e.g., passwords).
* Secure API endpoints with authentication and authorization.

**4.3** Usability

* User-friendly interfaces for web platforms.

**4.4** Scalability

* The system should support the addition of new features with minimal downtime.

**5. System Design**

**5.1** Architecture

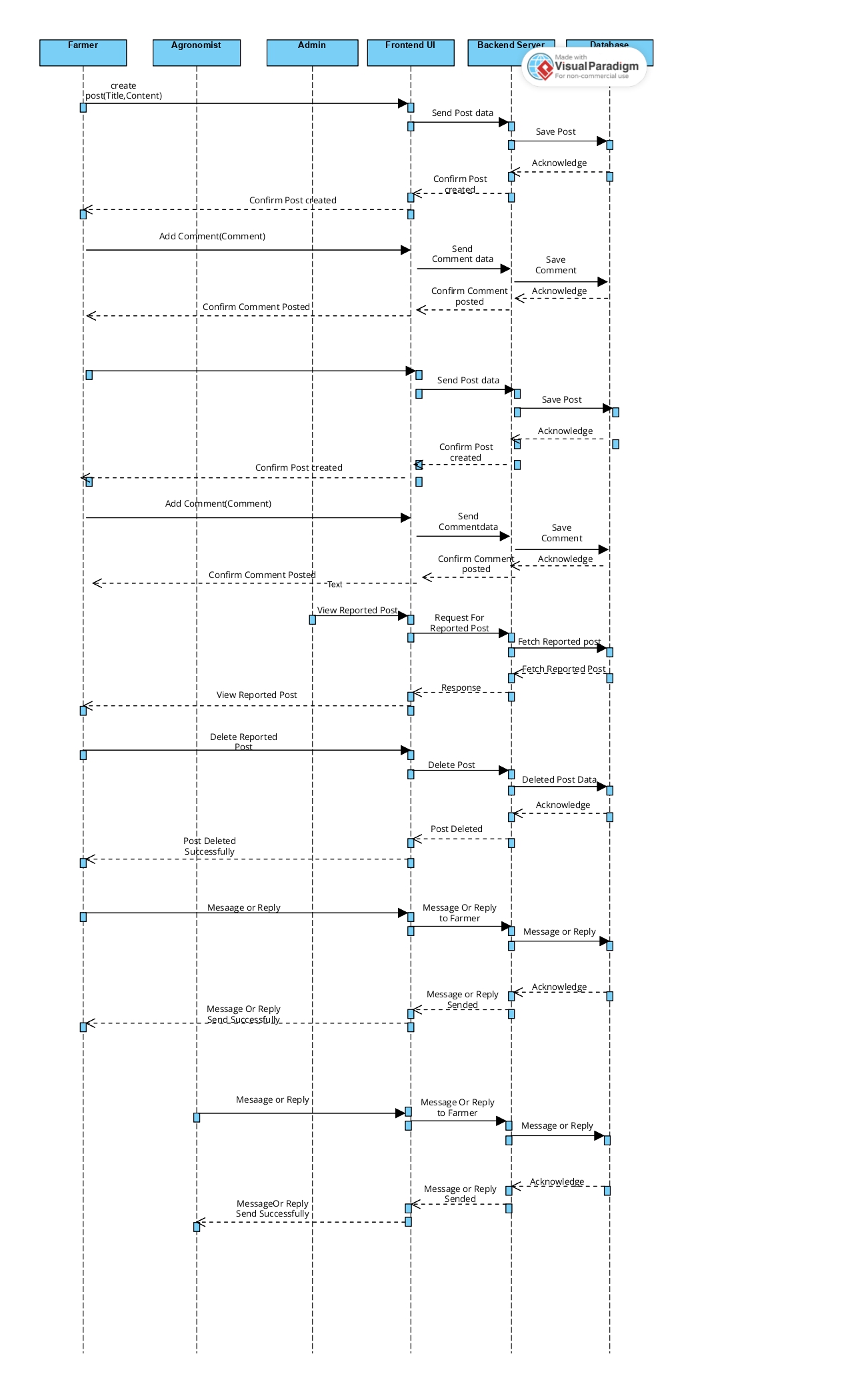
The system will follow a three-tier architecture:

1. **Presentation Layer**: React for web and mobile front-end.
2. **Business Logic Layer**: Spring Boot for backend services.
3. **Data Layer**: MySQL for database storage.

**5.2** Use Case Diagram



**5.3** Use Case Diagram:



* Goal of Implementation

The primary goal of implementing the Farmer Social Network is to bridge the gap between farmers and agronomists, enabling seamless communication and collaboration for improved agricultural practices. The system facilitates resource sharing, real-time chat, and post-based discussions, empowering users with expert advice and knowledge. It also ensures a secure, user-friendly, and scalable platform that promotes effective moderation and fosters community growth in the farming sector.